# Bradley **Kohler**

B.Eng. Mechatronics Engineering · 5 Years of Professional Work Experience

*Toronto, ON, Canada*

bradleykohler96@gmail.com | studentbrad | studentbrad

## Skills

| Programming Languages | Machine Learning | Microprocessor Firmware | Operating Systems | Hardware Design | Systems Design | Computer Mathematics | Continuous Integration |
|---|---|---|---|---|---|---|---|
| BASH | PyTorch | STM32 | Linux | Verilog HDL | AutoCAD | Maple18 | Docker |
| C/C++/C# | TensorFlow | Nordic | AndroidOS | NI Multisim | SolidWorks | NumPy | TravisCI |
| Python | | Qualcomm | mbedOS | Eagle | Maplesim | Eigen | CircleCI |
| Java | | Simulink | ChibiOS | | | MATLAB | Jenkins |
| HTML/CSS/JavaScript | | | FreeRTOS | | | | Bamboo |

## Work Experience

### STMicroelectronics — *Waterloo, Ontario*

Wireless Software Developer — *Sept. 2021 - Present*

- Developed the Radio Resource Control (RRC) Layer for LTE Cat M1/MTC/NB E-UTRA (3GPP specification 36.331) in embedded C.
- Debugged protocol stack issues in LTE layers RRC, PDCP, RLC, MAC, and L1 using Amarisoft and Rohde & Schwarz wireless communication conformance testing equipment.
- Developed the Application Layer (APL) for Zigbee, Thread, BLE, and IEEE 802.15.4 in embedded C including Over-The-Air (OTA), secure boot, and radio concurrency.
- Debugged protocol stack issues in Zigbee layers ZCL, ZDO, APS, NWK, and MAC using internally developed test runner.
- Generated and presented solutions to a larger group of developers to make quality improvements.

### Labforge Inc. — *Waterloo, Ontario*

Software & Firmware Developer — *May 2020 - Sept. 2021*

- Developed machine learning neural network structures, criteria, and optimization techniques; demonstrating good performance in the field.
- Designed and programmed new approaches to object re-identification and tracking in C/C++ and Python achieving fast results (100ms pipeline).
- Improved inertial sensor code bases in C/C++; running sensor processes as Unix systemd daemons on stereo cameras.
- Contributed in a corroborative effort with a team of software developers to a reliable C/C++ state estimation engine for stereo camera tracking.
- Communicated design ideas and coded with another software developer to create a robust C/C++ camera calibration software.

### Northern Digital Inc. — *Waterloo, Ontario*

Advanced Researcher & Firmware Developer — *May 2018 - Sept. 2019*

- Utilized mathematics skills to successfully design and program multiple data fusion algorithms in C/C++ and Python for 3D guidance systems (achieving NASA level TRL4) with real-time performance on offline systems (1-10ms pipeline).
- Worked collaboratively with a team of software developers to develop a fast C/C++ simulator (<10s) for a virtual reality headset/handremotes.
- Worked on custom hardware writing low-level firmware for sensors/peripherals including IMUs, ADCs, DACs, FLASH, UART, etc.
- Showed responsibility by coding CI/CD unit testing and deployment scripts for production products; automating testing using Bamboo/Jenkins.

### McMaster University — *Hamilton, Ontario*

Advanced Researcher — *May - Dec. 2015 & May - Sept. 2017*

- Worked with a team of software engineers developing software for safety critcial systems using Matlab Simulink.
- Successfully designed and built a prototype pacemaker using the Freescale K64F + custom PCB.

## Education

### McMaster University — *Hamilton, Ontario*

Mechatronics Engineering Co-op — *Sept. 2014 - April 2020*

- McMaster Cumulative Grade Point Average 3.7/4.0
- McMaster Engineering Co-op Student of the Year Nominee

## Projects

### Neural Networks for Wireless Transmissions — *Waterloo, Ontario*

Researcher & Developer — *October 2022 - May 2023*

- Developed several neural network models alongside a PhD graduate in artificial intelligence to detect anomalies in wireless air transmissions and stack protocol procedures.
- Used models such as RNN, RNN Attention-Based, and Transformer to detect anomalies in the nightly runs.

### Bottlenose — *Waterloo, Ontario*

Developer — *May 2020 - May 2021*

- Authored solutions for detection, re-identification, tracking and estimating past, present and future states of known objects.
- Primarily coded in Python and C/C++ using popular computer vision libraries such as PyTorch, GTSAM, OpenCV, etc.